GUI QT Beadandó

Potyogós amőba

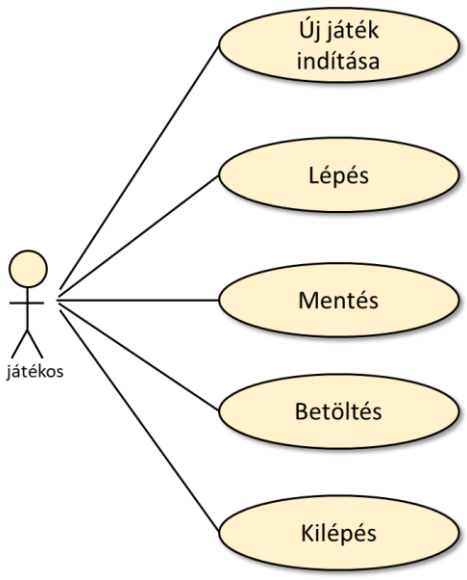
Rimóczi Loránd EOH12I

Feladat

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy $n \times m$ mezőből álló tábla (n az oszlopok, m a sorok száma), amelyre a játékosok O, illetve X jeleket potyogtatnak (azaz egy adott oszlopban a karakter mindig „leesik" a legalsó üres sorba, függetlenül attól, melyik sorban helyeztük le).A játékosok felváltva lépnek, és egy oszlopban csak akkor helyezhetnek el új jelet, ha az még nem telt meg. A játékot az nyeri, aki előbb elhelyez vízszintesen, függőlegesen, vagy átlósan négy szomszédos jelet. A játék döntetlennel ér véget, ha betelik a tábla. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (8×5, 10×6, 12×7), valamint az aktuális játék mentésére és egy korábban elmentett játék betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött, illetve azt is, ha döntetlen lett a vége, majd automatikusan kezdjen új játékot.

A feladat elemzése

A játékban két játékos vesz részt, de az alkalmazás szempontjából mindig csak egy felhasználó van, az egyetlen különbség a potyogó jel formája. Ő ötféle tevékenységet végezhet.



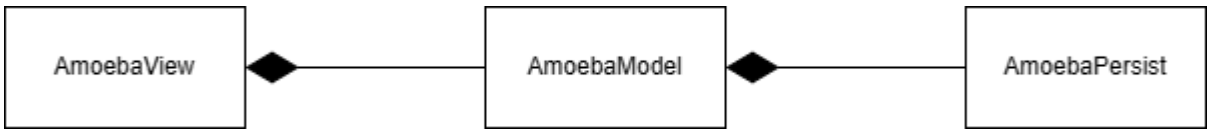
A felhasználói tevékenységek során az alábbi esetek következhetnek be.

1	Alkalmazás indítása	GIVEN:	Az alkalmazás telepítve van.
		WHEN:	Alkalmazás indítása.
		THEN:	Megjelenik az üres játéktábla.
2	Kilépés	GIVEN:	A játéktábla aktív.
		WHEN:	A játékfelület ablakának lezáró ikonjára kattintunk.
		THEN:	Az alkalmazás bezáródik.
3a	Lépés	GIVEN:	A játéktábla aktív.
		WHEN:	A tábla feletti gombra kattintva kiválasztjuk, hogy melyik oszlopba akarjuk tenni a jelölő karaktert.
		THEN:	Attól függően, hogy ki a soron következő játékos egy ’X’ vagy ’O’ jel kerül a kiválasztott oszlopba.
3b	Lépés	GIVEN:	A játéktábla aktív.
		WHEN:	Olyan oszlop fölötti gombra kattintunk, ahol van még üres mező, de ezzel betelik a tábla.
		THEN:	Attól függően, hogy ki a soron következő játékos egy ’X’ vagy ’O’ jel kerül a kiválasztott oszlopba, majd egy döntetlen üzenet jelenik meg. Ennek elolvasása után új játék kezdődik.
3c	Lépés	GIVEN:	A játéktábla aktív.
		WHEN:	Üres mezőt tartalmazó oszlop fölötti gombra kattintunk, és ezzel négy egyforma jel jelenik meg egy sorban, oszlopban vagy átlóban.
		THEN:	Attól függően, hogy ki a soron következő játékos egy ’X’ vagy ’O’ jel kerül a kiválasztott oszlopba, majd egy győzelmi üzenet érkezik a megfelelő karakter jellel. Ennek elolvasása után új játék kezdődik.
3d	Lépés	GIVEN:	A játéktábla aktív.
		WHEN:	Olyan oszlop fölötti gombra kattintunk, ahol nincs több üres mező.
		THEN:	Nem változik a játék állapota, és újra ugyan az a játékos jön.

4a	Új játék	GIVEN:	A játéktábla aktív.
		WHEN:	A New: 8 x 5 gombra kattintunk.
		THEN:	Megjelenik az üres játéktábla 8 x 5-ös méretben.
4b	Új játék	GIVEN:	A játéktábla aktív.
		WHEN:	A New: 10 x 6 gombra kattintunk.
		THEN:	Megjelenik az üres játéktábla 10 x 6-os méretben.
4c	Mentés	GIVEN:	A játéktábla aktív.
		WHEN:	A New: 12 x 7 gombra kattintunk.
		THEN:	Megjelenik az üres játéktábla 12 x 7-es méretben.
5	Mentés	GIVEN:	A játéktábla aktív.
		WHEN:	A Save gombra kattintunk.
		THEN:	Mentés név bekérése, majd a mentés végrehajtása
6	Betöltés	GIVEN:	A játéktábla aktív.
		WHEN:	A Load gombra kattintunk.
		THEN:	Betöltődik egy mentés kiválasztó fájlböngésző felület, megfelelőt kiválasztva a mentés betöltődik.

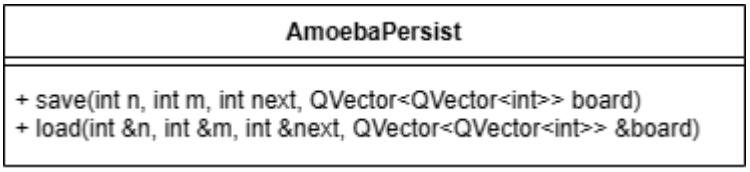
Architektúra

Az alkalmazás három rétegű (view-model-presistence) architektúrában valósult meg. A view (nézet) a játéktáblát megjelenítő réteg, a model felel a program logikájáért, míg a persistence (perzisztencia) az adattárolásért, azaz a fájlba való mentésért.



Perzisztencia

A korábban elmentett játékállás egy .save file-ban tárolódik. Az persistence réteg egyetlen osztálya ezen állományokhoz történő hozzáférést biztosítja a modell számára úgy, hogy a modellnek nem kell tudnia arról, hogy milyen módon történik a tárolás.



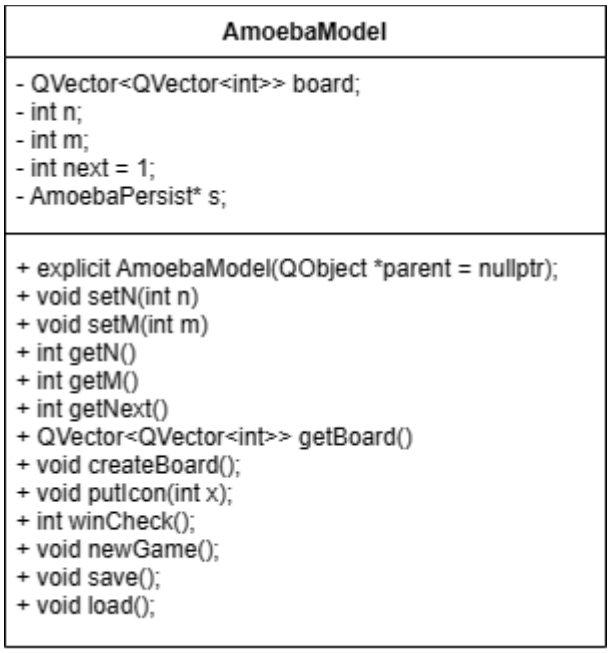
A persistence réteg kettő metódussal rendelkezik:

A save() elmenti az aktuális „meccset”, azaz a tábla méretét, hogy ki következik, valamint magának a táblának az állapotát, egy bekért név.save nevű szöveges fájlba.

A load() betölti a save() által mentett adatokat a kiválasztott mentés.save fájlból, majd annak megfelelően újra rajzolja a felületet.

Model

A model tárolja a futás közbeni állapotot (tábla mezőin a karakterek, ki következik), valamint kezeli a játék logikáját, mint például győzelem ellenőrzés. A model publikus metódusait a nézet hívja meg, míg a másik irányba két signal küld jelet a megfelelő metódusok futtatásáért.



Adattagok:

Az n és az m a tábla méretét, a next a következő játékost, a board pedig magát a táblát tárolja el.

A privát adattagok értéke (tábla mérete, következő játékos, tábla tartalma) a nekik megfelelő getter metódusokkal lekérdezhető, a setter metódusokkal pedig beállítható.

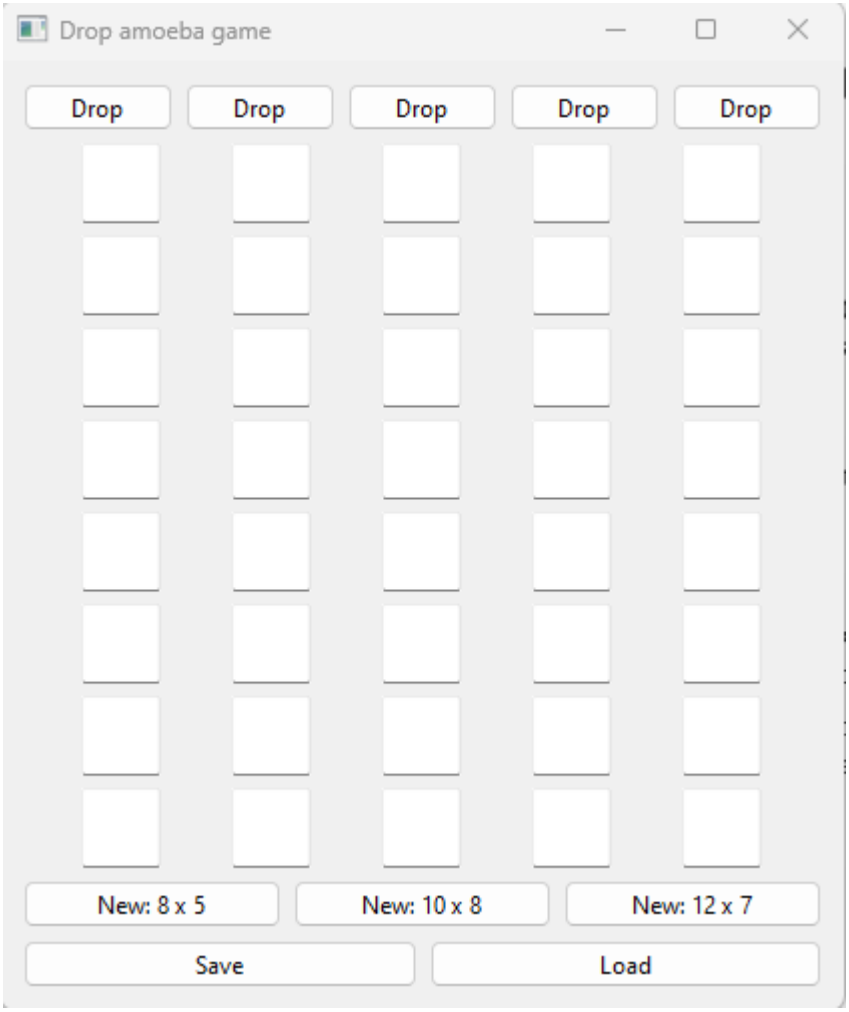
A createBoard() egy új játéktábla generálását, a putItem(x) a játék egy lépését végzi. A putItem(x) minden lépésnél lefuttatja a winCheck() metódust, hogy megvizsgálja, az adott lépéssel kialakult-e győztes helyzet.

Az új játékot a newGame() metódus generálja, a load() és save() metódusok pedig a perzisztenciánál látott metódusokat hívják a mentéshez, betöltéshez.

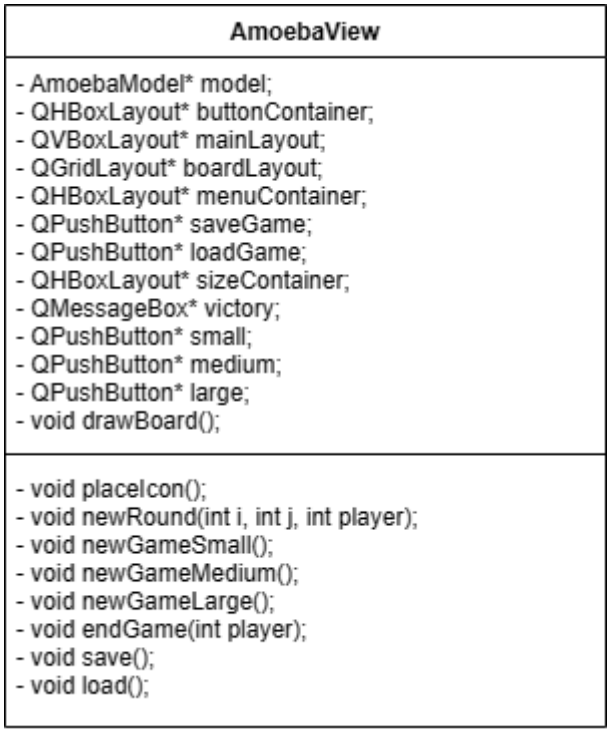
View

A view (nézet) réteg felel a játék képernyőn történő megjelenítésért. A megjelenítés megoldása egyszerű, gombokból és QLineEdit által reprezentált játék mezőkből áll.

A játék felületén a játék mezők feletti Drop gombokkal lehet potyogtatni, a mezők alatti részen vannak az új játék indításához szükséges gombok, alattuk pedig a mentés (Save) és betöltés (Load) gombok.



Osztálydiagram:



Vezérlők és adattagok:

A főablak osztályának adattagjai a játék megjelenítésére szolgáló grafikai elemeket (tábla, játékosok jelei) tárolják, valamint egy-egy hivatkozást a mentés és betöltéshez.

Metódusai kizárólag a megjelenítést, kirajzolást oldják meg.

Eseménykezelés:

Sender	Signal	Reciever	Slot
model	updateBoard(int, int, int)	AmoebaView	newRound(int, int, int)
model	gameOver(int)	AmoebaView	endGame(int)
saveGame	clicked()	AmoebaView	save()
loadGame	clicked()	AmoebaView	load()

Végfelhasználói tesztesetek:

1	Az alkalmazás indítása.	Megjelenik a játék programablak.
2	Kilépés folyó játékból.	Az alkalmazás leáll.
3a	Váltakozó lépések.	Az oszlop feletti gombokra kattintva váltakozva hol ’X’, hol ’O’ jelenik meg a mezőn.
3b	Nincs hely az oszlopban.	Nincs változás a táblán.
3c	Négy ugyanolyan jel egy sorban, oszlopban vagy átlóban.	Győztes megjelnítése, új játék indítása.
3d	Elfogynak az üres mezők, de nincs nyertes.	Megjelenik a döntetlen üzenet, majd új játék kezdődik.
4a	Új játék indítása egy folyamatban levő játéknál	Megjelenik az üres játéktábla.
4b	Új játék indítása egy befejeződött játéknál.	Megjelenik az üres játéktábla.
5	Mentés egy futó játékról.	A program elmenti az aktuális játékállást, ami után folytatni lehet a játékot.
6	Betöltés üres/folyamatban lévő játéknál.	Az elmentett játék betöltődik, azzal lehet tovább játszani.